### **Introducing Remote Shell Control**

Remote Shell Control lets you send user-defined commands to a remote host using either the RSH protocol or the REXEC protocol.

The rshctrl.ocx establishes a connection between the client machine and the server machine. Commands sent by the client machine through this connection can be executed on the server machine. Remote Shell Control returns the data to the client machine.

**Note:** The client machine must have TCP/IP installed correctly because the RshCtrl is TCP/IP network-based. This also forces the control to use events to notify the client machine when the server has sent something back.

You can embed the control anywhere without worrying about its appearance because the control is invisible at runtime.

You need to supply the following properties before any client/server transaction can take place:

HostName	The name or IP address of the server machine.
RemoteUsername	The user account name to use on the server machine.
Command	The command to be sent.
RemotePassword (REXEC)	The remote password (SENT IN CLEARTEXT).
LocalUsername (RSH)	The username on the client machine.

**Note:** Some required information depends upon the protocol you are using. For example, if you are using the REXEC protocol, you need the remote password. If you are using the RSH protocol, you need the local user name.

Use the AsyncStartCmd method after completing the required properties. At this point, a connection is established and the command is sent. As data returns from the server, the DataReady Event is generated.

# **ErrorMsg**

Returns a string containing the last error message. You can change this property only when there are no outstanding commands.

Syntax	[Form.]RshCtrl.ErrorMsg
Parameters	This property has no parameters.
Data Type	String (read only)

#### **HostName**

The remote host to which to connect. The HostName can either be an IP address or a server name specified in the host table. You can change this property only when there are no outstanding commands.

Syntax	<pre>[Form.] RshCtrl.HostName[ = nameString\$]</pre>
Parameters	This property has no parameters.
Data Type	String (read only)

**Note:** If the host name given is invalid, an error string returns in both ErrorMsg and in the returning data string.

### LocalUsername

The local user name identifies you on the current machine. You can change this property only when there are no outstanding commands. Use this property only when you set ProtocolUsed Property to RSH.

Syntax	[Form.] RshCtrl.LocalUsername[ =nameString\$]
Parameters	This property has no parameters.
Data Type	String

## Command

The command to send to the host. You can change this property only when there are no outstanding commands.

Syntax	<pre>[Form.] RshCtrl.Command[ = cmdString\$]</pre>
Parameters	This property has no parameters.
Data Type	String

## RemoteUsername

The user name of your server account to use to login. You can change this property only when there are no outstanding commands.

Syntax	[Form.] RshCtrl.RemoteUsername[=nameString\$]
Parameters	This property has no parameters.
Data Type	String

#### RemotePassword

The password for your remote user name to use for login. You can change this property only when there are no outstanding commands. Use this property only when you set ProtocolUsed Property to REXEC.

Syntax	[Form.] RshCtrl.RemotePassword[ =pwdString\$]
Parameters	This property has no parameters.
Data Type	String

**Note:** The password transfers in cleartext. Do not use this command over the public Internet because malicious users can get any user name and password combination sent via this command.

## long ProtocolUsed

The protocol that sends the command. You can change this property only when there are no outstanding commands.

The setting of this property effects the properties required for a command.

Syntax	[Form.] RshCtrl.ProtocolUsed[ =PROTOCOLTYPE]
Parameters	PROTOCOLTYPE—the property for the protocol you want to use:  To use the RSH protocol, set this property to PROTOCOL_RSH.  To use the REXEC protocol, set this property to PROTOCOL_REXEC.
Data Type	Long

### **SocketLogging**

This turns socket logging on (true) or off (false). You can change this property only when there are no outstanding commands.

The socket log contains all socket connection information in a format that Hummingbird Technical Support personnel can read. You can also use SocketLogging to see exactly what information the control sends and receives.

Syntax	[Form.] RshCtrl.SocketLogging[ = {True False}]
Parameters	This property has no parameters.
Data Type	Boolean

**Note:** If you are going to change the Socket logging path to something other than the default, you must change the path before enabling this property. If you do not, the socket path change has no effect.

#### **Related Topics**

**ErrorMsg** 

**LoggingPath** 

## LoggingPath

The file to which to write the socket information. You can change this property only when there are no outstanding commands.

Syntax	[Form.] RshCtrl.LoggingPath[=filepathString\$]
Parameters	This property has no parameters.
Data Type	String

Note: The default LoggingPath is rshdbg.txt, which resides in the same directory as rshctrl.ocx. You must change this path before enabling the SocketLogging property if you are going to change the Socket logging path to something other than the default. If you do not, this path change has no effect.

#### **Related Topics**

ErrorMsg SocketLogging

## **DataReadyEvent**

When the control receives a packet of data, The DataReadyEvent is generated.

Syntax	Private Sub RshCtrl_OnDataReadyEvent ()
Return Value	None

Note: The DataReadyEvent signifies that the GetData Method now returns at least one packet of data. More than one packet may be available when the DataReadyEvent is generated.

## **RCmdError**

This event is used when communications between the client and the host have terminated prematurely.

Syntax	Private Sub RshCtrl_OnRCmdError(LONG, STRING)
Parameters	error code—helps determine problems. (LONG) error msg—describes the error condition. (STRING)
Return Value	None

**Note:** Usually, this command is used if there is a socket problem or if the host is unreachable.

## **RCmdCompletedEvent**

When the server terminates the connection between the client and the server, the RCmdCompletedEvent is generated. The RcmdCompletedEvent is also generated when the hostname given resolves to an unknown IP address or the connection is not established.

Syntax	Private Sub RshCtrl_OnRCmdCompletedEvent	
Return Value	None	

**Note:** The RCmdCompletedEvent signifies that the previous outstanding Remote command process has ended. Properties that are not allowed to change during the remote command process can now be changed again.

## **AsyncSendRCmd**

#### Starts the Remote Command process.

Syntax	Declare Function AsyncSendRCmd () As Boolean	
Parameters	This method has no parameters	
Return Value	<ul> <li>True, if all the parameters (except for the host name) are valid.</li> <li>False, if a required parameter is missing or invalid. The ErrorMsg Property provides the reason.</li> </ul>	

**Note:** This is an asynchronous command. It returns immediately with no guarantee the connection was successful. All results come from Events.

#### **Related Topic**

**ErrorMsg** 

## CancelAsyncRCmd

#### Stops the outstanding Remote Command process.

Syntax	Declare Sub CancelAsyncRCmd ()	
Parameters	This method has no parameters	
Return Value	None	

**Note:** This command works only when there are outstanding commands. Otherwise, the connection between the client and the server ends. Existing data remains in the Data Queue waiting for subsequent GetData calls.

## **GetData** [out]

Get a packet of previously received data that has been received from the host.

Syntax	Declare Function GetData (ByRef retStr As Variant [OUT]) As Boolean <cr></cr>	
Parameters	RefStr—Reference to Buffer to hold the incoming string. (VARIANT)	
Return Value	<ul> <li>True, if data copies into the outgoing parameter</li> <li>False, if no data is currently available</li> </ul>	

**Note:** This function returns, via the refStr parameter, the first packet received and is waiting in the RshCtrl's data queue.

# **AboutBox**

When used, this method launches a dialog box containing version information of rshctrl.ocx.

Syntax	Declare Sub AboutBox ()	
Parameters	This method has no parameters.	
Return Value	None	

#### **General Accessibility**

Hummingbird products are accessible to all users. Wherever possible, our software adheres to Microsoft Windows interface standards and contains a comprehensive set of accessibility features.

Access Keys All menus have associated access keys (mnemonics) that let you use the keyboard, rather than a mouse, to navigate the user interface (UI). These access keys appear as underlined letters in the names of most UI items. (If this is not the case, press Alt to reveal them.) To open any menu, press Alt and then press the key that corresponds with the underlined letter in the menu name. For example, to access the File menu in any Hummingbird application, press Alt+F.

Once you have opened a menu, you can access an item on the menu by pressing the underlined letter in the menu item name, or you can use the arrow keys to navigate the menu list.

**Keyboard Shortcuts** Some often-used menu options also have shortcut (accelerator) keys. The shortcut key for an item appears beside it on the menu.

**Directional Arrows** Use the directional arrows on the keyboard to navigate through menu items or to scroll vertically and horizontally. You can also use the directional arrows to navigate through multiple options. For example, if you have a series of radio buttons, you can use the arrow keys to navigate the possible selections.

**Tab Key Sequence** To navigate through a dialog box, press the Tab key. Selected items appear with a dotted border. You can also press Shift+Tab to go back to a previous selection within the dialog box.

**Spacebar** Press the Spacebar to select or clear check boxes, or to select buttons in a dialog box.

**Esc** Press the Esc key to close a dialog box without implementing any new settings.

**Enter** Press the Enter key to select the highlighted item or to close a dialog box and apply the new settings. You can also press the Enter key to close all About boxes.

**ToolTips** ToolTips appear for all functional icons. This feature lets users use Screen Reviewers to make interface information available through synthesized speech or through a refreshable Braille display.

### **Microsoft Accessibility Options**

Microsoft Windows environments contain accessibility options that let you change how you interact with the software. These options can add sound, increase the magnification, and create sticky keys.

#### To enable/disable Accessibility options:

- 1. In Control Panel, double-click Accessibility Options.
- 2. In the Accessibility Options dialog box, select or clear the option check boxes on the various tabs as required, and click Apply.
- 3. Click OK.

If you installed the Microsoft Accessibility components for your Windows system, you can find additional accessibility tools under Accessibility on the Start menu.

# **Technical Support**

You can contact the Hummingbird Technical Support department Monday to Friday between 8:00 a.m. and 8:00 p.m. Eastern Time.

Hummingbird Ltd. 1 Sparks Avenue, Toronto, Ontario, Canada M2H 2W1					
	Canada and the USA	International			
Technical Support:	1-800-486-0095	+1-416-496-2200			
General Enquiry:	1-877-FLY-HUMM	+1-416-496-2200			
Main:	+1-416-496-2200				
Fax:	x: +1-416-496-2207				
E-mail:	support@hummingbird.com				
FTP:	ftp.hummingbird.com				
Web Support:	support.hummingbird.com/customer				
Web Site:	www.hummingbird.com				